

Réalité augmentée

Marqueur Image

Html5/JS

Christophe Vestri

Le mardi 17 janvier 2019

Objectifs du cours

- Connaitre/approfondir la RA
- Avoir quelques bases théoriques
- Expérimenter quelques méthodes et outils
- Réaliser un projet en RA
- Evaluation:
 - Présence (20%)
 - Participation en classe (40%)
 - Projet (40%)

Plan du cours

- 8 janvier : Réalité augmentée intro, Unity/Vuforia et projet final
 - 17 janvier: Construction RA + Notions 2D + exercices JS
 - 22 janvier: QRCode et Unity/vuforia et/ou projet
 - 29 janvier: Vision par ordinateur et RA (openCV – exo C++)
 - 5 février : Résumé et présentation des Projets
-
- **Suite: Cours Cartographie/JS/AR/VR**

Plan Cours 2

- Rappel
- Systèmes RA
- Théorie Image
- Projet ArtMobilib
 - Démonstration ArtMobilis
 - 3 librairies JavaScript et 3 exercices
- Projet Final
 - Développement d'une démo Start wars

Rappel du premier cours

Qu'est-ce que la Réalité augmentée?

- Augmentée:
 - Amplifier
 - Rehausser
 - Améliorer
- [Wikipédia](#): La **réalité augmentée** désigne les systèmes informatiques qui rendent possible la superposition d'un modèle virtuel 2D ou 3D à la perception que nous avons naturellement de la réalité et ceci en temps réel.
- [RAPro](#) : Combiner le monde réel et des données virtuelles en temps réel

Autre définition de la RA

- [RAPro](#) : Combiner le monde réel et des données virtuelles en temps réel
- 5 sens:
 - Visuel: smartphone, lunettes...
 - Sonore: déficients visuels
 - Tactile/haptique: systèmes retour de force
 - Odorat: Cinema 4D
 - Goût:

Questions Cruciales

Antoine Morice
ISM Marseille

Santé & sécurité : Sécurité (e.g., occlusion du champ visuel par les dispositif de visualisation tête haute, TMS causés par le port de VTH) et santé de l'employé (e.g., ophtalmie et impact de la lumière bleue sur la rétine, problèmes d'accommodation, cyber-malaises, etc.)

Ethiques & Juridiques : Informatique et liberté (e.g., monitoring permanent de l'activité), big data (e.g., conservation de données personnelles sur le comportement de l'opérateur, les regards, CDU d'Oculus Rift autorisent la firme à collecter des informations sur les mouvements physiques des utilisateurs, etc.)

Techniques : Technologie utilisée (e.g., visiocasque vs. projection, géolocalisation, etc.), modèles et maquettes numériques (e.g., inventaires, scan 3D, réalisme des modèles, précision et résolution, etc.

Ergonomiques : adaptation à tous les secteurs, à tous les publics de l'industrie (e.g., standards, anthropométrie, etc.), aux environnements (e.g., luminosité, thermmie)

Managériales et commerciales : conduite du changement, formation des personnels, ordre d'introduction dans les différents secteurs de l'entreprise, intégration de la réalité augmentée dans les missions des prestataires ou des fournisseurs

Comment choisir son système de RA

Antoine Morice
ISM Marseille

- Systèmes (displays, tracking)



Tête portés



Tenus



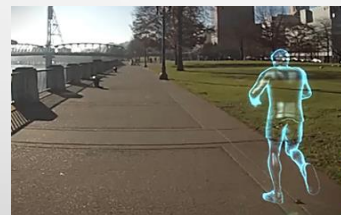
Spatiaux

- Utilisations, fonctions, objectifs

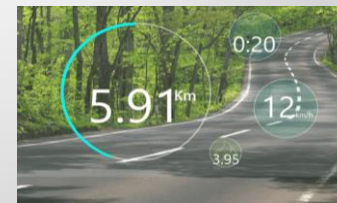


Amélioration de la Pratique Compréhension du Spectacle

- Modalités de présentation



Avatar



Icones & texte



Courbes & jauges

- Type (addition, translation, amélioration, ...)

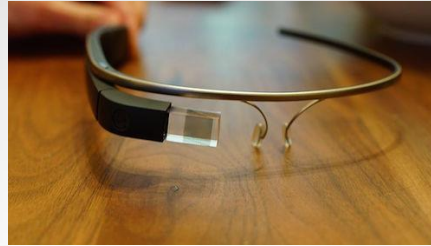
Principaux systèmes de RA

3 types d'affichage:

- Via un dispositif semi-transparent
- Par projection
- Affichage sur flux vidéo

Lunettes de RA

- Google



Glass



Glass 2 (A4R-GG1)

- VuViZ



M300

- Daqri



Smart Helmet

- Laster



WAVE

Lunettes de RA/RV/RM

- Hololens



- Magic Leap One



Affichage par projection

Pranav Mistry - [SixthSense](#)



Principaux systèmes de RA

Affichage sur flux vidéo, caméra ou smartphone,
2 systèmes:

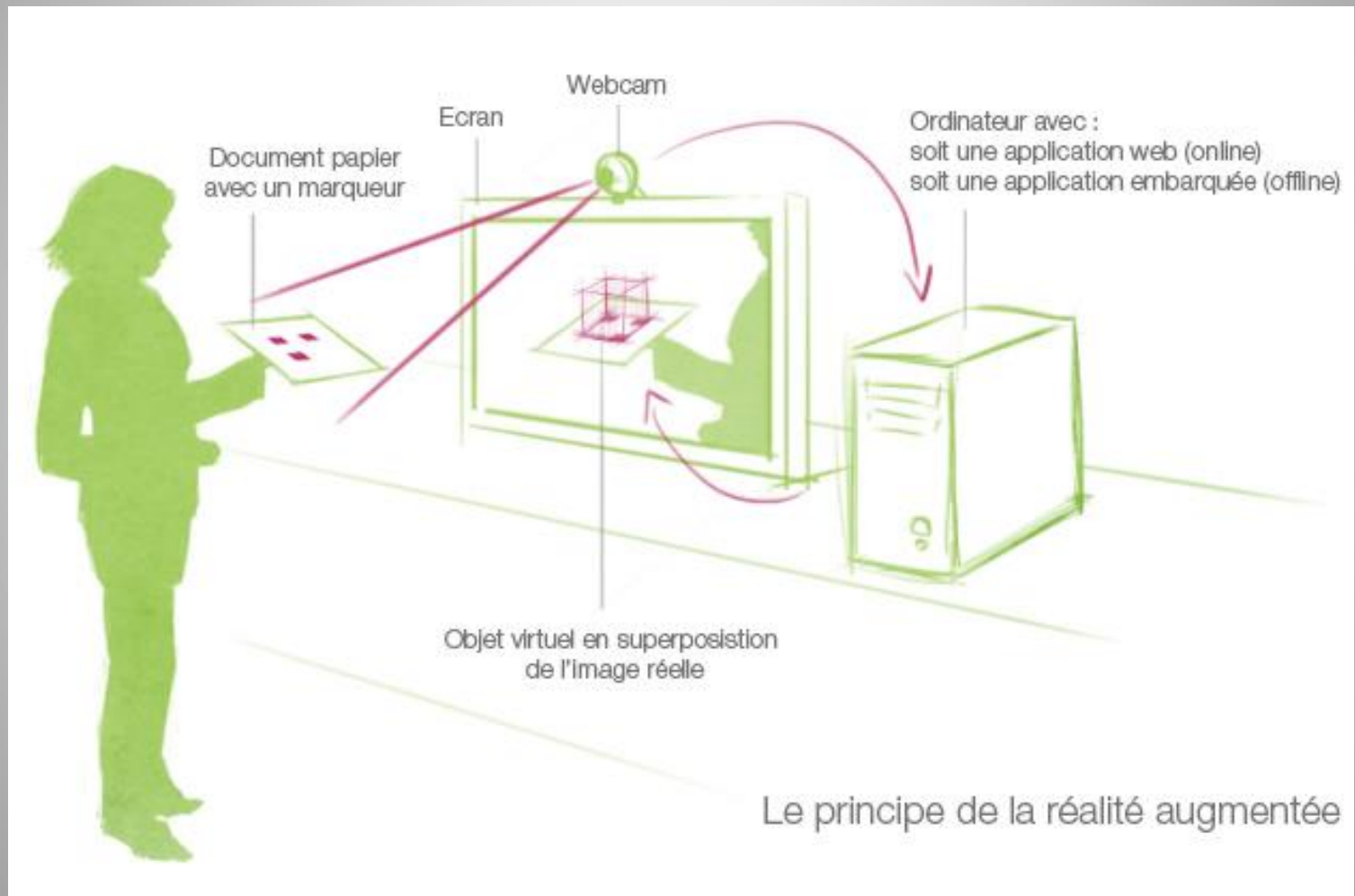
- RA avec caméra fixe
- RA Mobile: la caméra est en mouvement

Principaux systèmes de RA

Affichage sur flux vidéo, caméra ou smartphone,
2 systèmes:

- RA avec caméra fixe
- RA Mobile: la caméra est en mouvement

RA Fixe



RA avec caméra Fixe



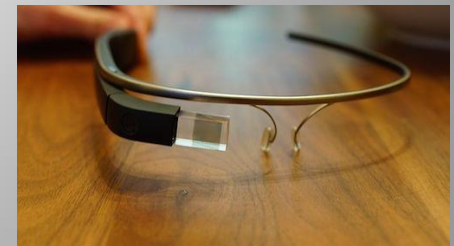
Démo
National
Geographics

Magic Mirror



RA avec caméra Mobile

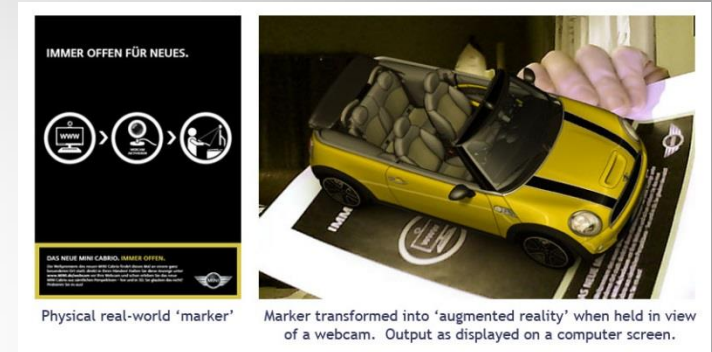
- Smartphones, tout pour la RA
 - Camera + écran – déterminer/montrer ce qui doit être vu
 - Donnée GPS– localisation
 - Compas – quelle direction on regarde
 - Accéléromètre – orientation
 - Connection Internet – fournir des données utiles
- 58% des Français ont un smartphone en 2015
- 90% des 18-24ans
- Lunettes de RA et VR



Types de RA mobile

Marqueurs:

- Caméra pour détecter un marqueur dans le monde réel
- Calcul de sa position et orientation
- Augmente la réalité



Géolocalisation:

- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité



Types de RA mobile

Utilisation de marqueurs caméras:

- Marqueurs Spécifiques:
 - Tag visuels
 - Formes spécifiques (carrés, cercles)
- Marqueurs Images
 - Photo, image de l'objet/scène
- Processus de RA
 - Détection du marqueur dans la vidéo
 - Transformation 2D-3D
 - Affichage 3D



Exemple de Marqueur image



- Pour faire de la RA, il va falloir
 - Retrouver l'image,
 - la délimiter
 - Dans toutes les conditions (proche, loin, oblique)

Exemple de Marqueur image

- Concrètement il va falloir
 - Avoir un moyen pour décrire l'image de référence



- Avoir un moyen de retrouver
- De le différencier des autres images

-> Vision par ordinateur

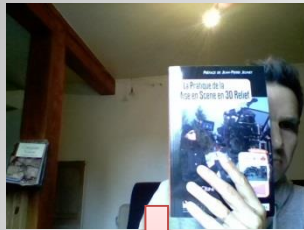


Vision par ordinateur et RA

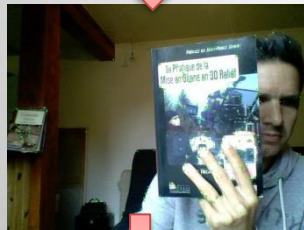
- Analyse image/vidéo -> vision par ordinateur
- Plusieurs technologies
 - Détection de marqueurs spécifiques: coins, primitives naturels, carrés, ronds
 - Mise en correspondance: primitives, images
 - Reconnaissance d'image: monument, façade, visage
 - Reconnaissance d'objets: tables, chaise....
 - Recalage caméra: calcule de la pose
 - Traitement d'image: contraste, segmentation
 - Mixer image et synthétique

Technologies nécessaires

Références Acquisition vidéo



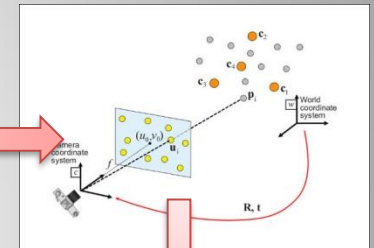
Détection coins et descripteurs



Matching

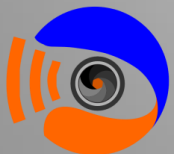
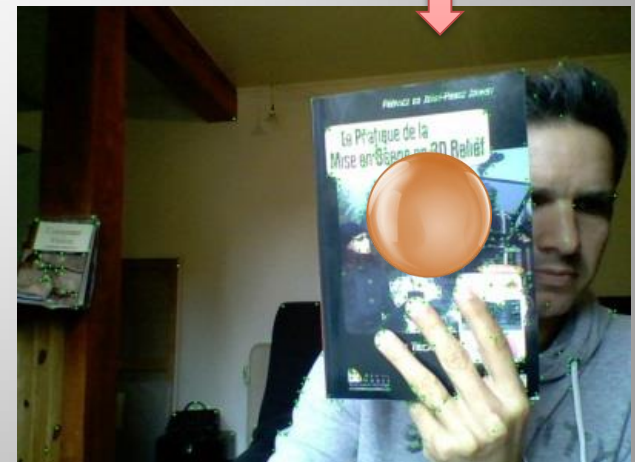
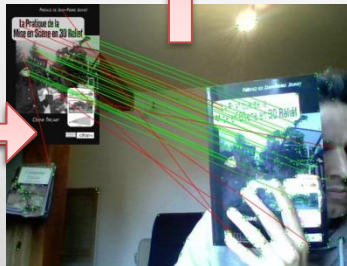


Calcul de la pose caméra



Affichage 3D

Localisation du Pattern



Detection et Appariement

- Plusieurs méthodes existent pour décrire, détecter, et appairer les images
- Pixels, points, segments, régions, et droites des images peuvent être utilisées
- Quatre étapes sont nécessaires dans la détection et l'appariement des primitives
 - Détection de primitives
 - Description des primitives
 - Appariement des primitives
 - Tracking de primitives

Quelques termes

- **Marqueur** - utilisé pour spécifier où et quelle information ou contenu doit être placé (spécifiques ou image)
- **Primitives naturelles** – points/parties d'un objet visualisé
- **Detecteur** – utilisé pour rechercher dans les images les points spécifiques répétitifs
- **Descripteur** – utilisé pour caractériser les points ou région à partir de l'image. Ils sont utilisés dans la mise en correspondance
- **Canal** – association d'un marqueur à l'objet synthétique à afficher

Qu'est-ce qu'une image

- Image de couleur = 3 images (+ alpha)
- Algorithmes avec 1 entrée => Image de gris

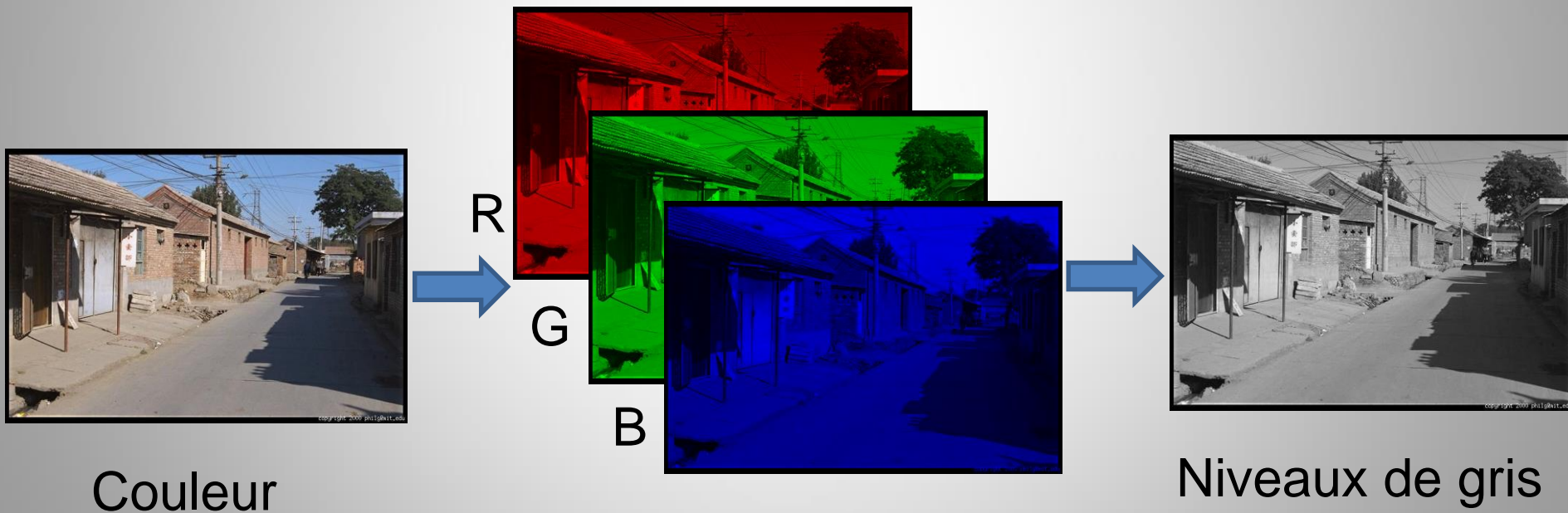


Image de float

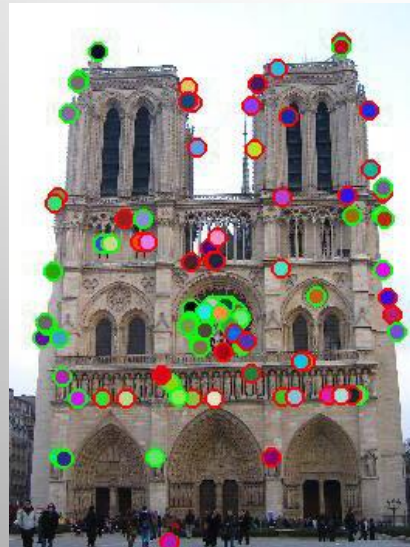


0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

000 philg@mit.edu

Qu'est-ce qu'une primitive

- Une primitive c'est:
- Un élément spécifique de l'image
- Pixels/Point/coin unique de l'image
- Utilisé pour représenter/simplifier l'information contenue dans l'image



Points



Qu'est-ce qu'une primitive

Ca peut être aussi



Régions



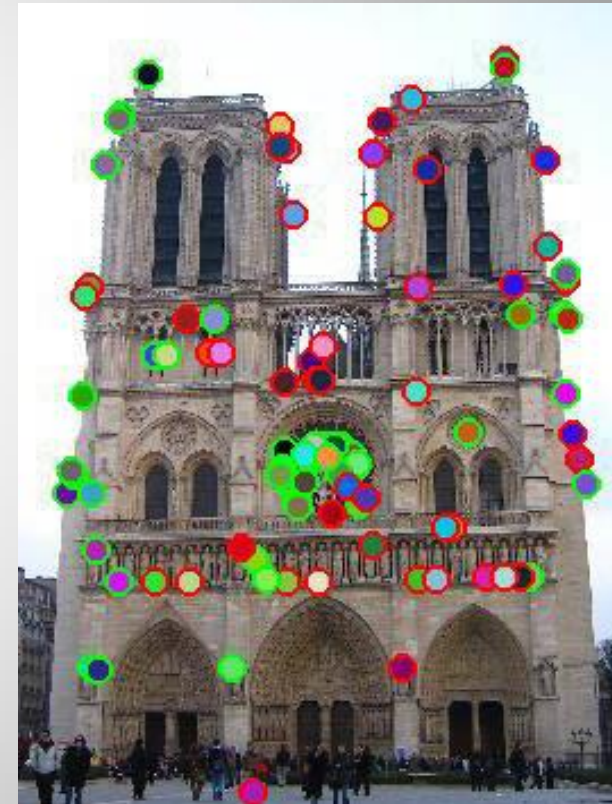
Segments

Contours



Détecteur de primitive

- Il va extraire/sélectionner les primitives de l'image
- Critères de qualité:
 - Caractérisables: distinctif, particularité, reconnaissable, précision
 - Répétabilité et invariance: échelle, rotation, illumination, point de vue, bruit



Détecteur de primitive

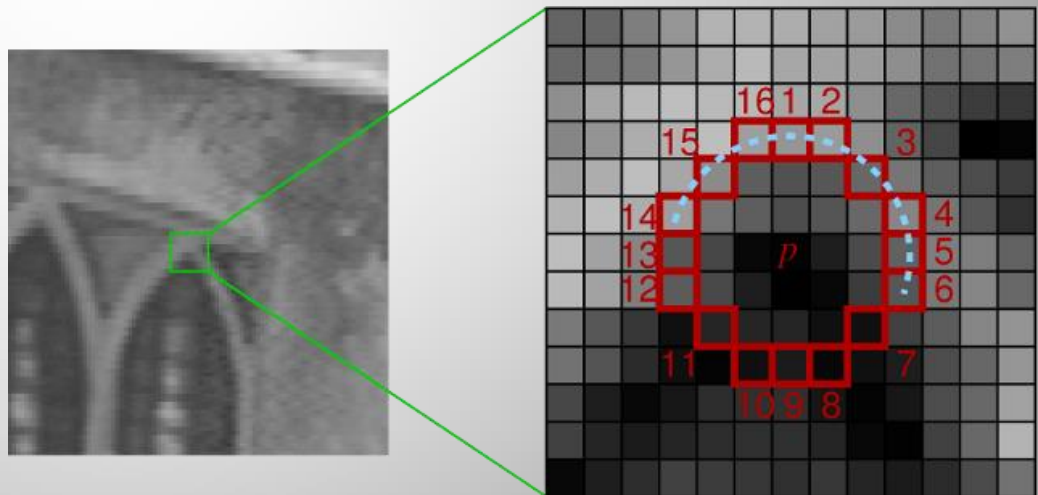


Détection de coins

FAST: Features from Accelerated Segment Test

<http://www.edwardrosten.com/work/fast.html>

- Cercle Bresenham 16 pixels autour du point analysé
- On détecte un coin en p si
l'intensité de N pixels
est $>$ ou $<$ de $X\%$ à I_p
- Rapide et robuste



Descripteur de points

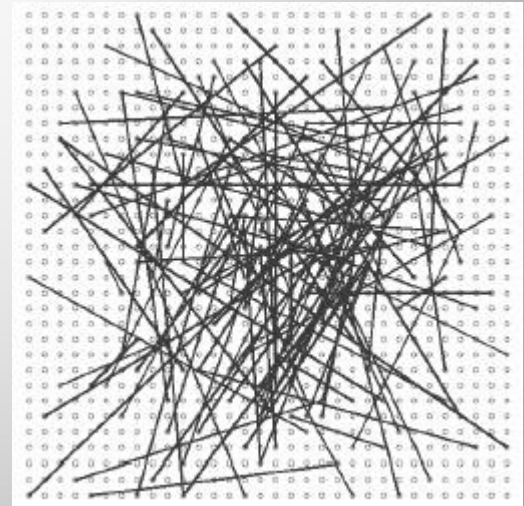
- Description du point à partir de l'image (locale)
- Utilisé pour l'appariement
 - Stockage des descriptions des marqueurs image
 - Comparer avec les primitives de l'image courante
- Critères de qualité:
 - Discriminant
 - Invariant : échelle, rotation, illumination, point de vue, bruit
 - Rapide et empreinte mémoire faible

Descripteur de points

BRIEF : Binary robust independent elementary features

<http://cvlab.epfl.ch/research/detect/brief>

- Vecteur de N paires de points sur un patch
- Comparaison pour chaque paire
 - Si $I_1 < I_2$ alors $c=1$
 - Sinon $c=0$
- Descripteur=100101001...
- Rapide et robuste

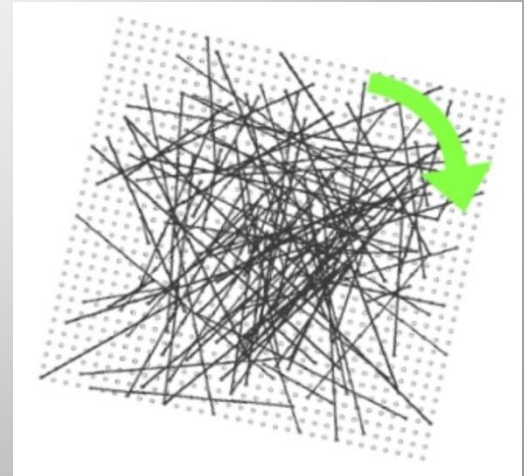
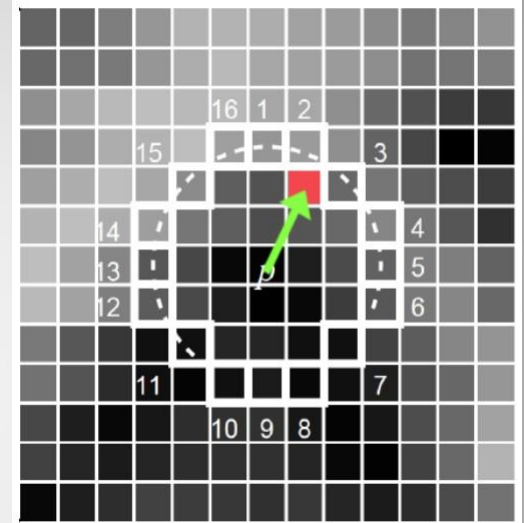


Descripteur de points

ORB (Oriented FAST and Rotated BRIEF)

http://docs.opencv.org/.../py_feature2d/py_orb/py_orb.html

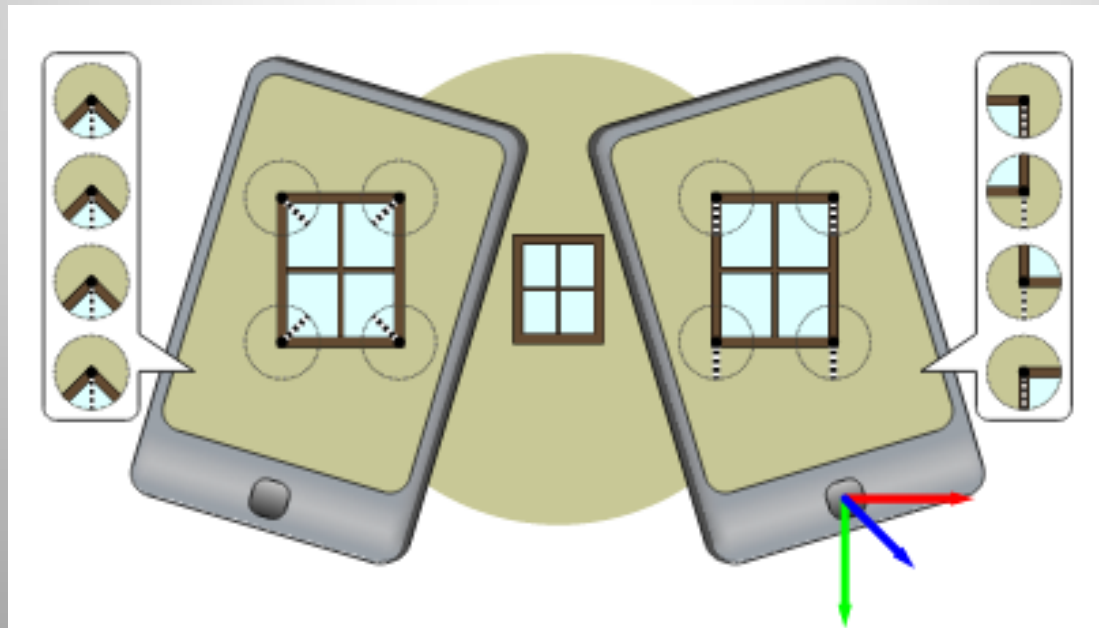
- Prise en compte rotation pour robustesse
- Direction=pixel avec variation la plus forte
- Rotated BRIEF pour aligner les descripteurs lors du matching



Descripteur de points

Autre exemple: GAFD Gravity Aligned Feature Descriptors

- Utilisé par Metaio (Apple)
- Utilise les capteur inertiel pour avoir des descripteurs alignés avec la gravité



Reconnaissance par matching

Appariement des coins

- Brute force matching, on teste toutes les paires
- Similarité= Distance de Hamming (nombre de bits différents)

A = 1 0 1 1 0 0 1 0 0 1 0 0

B = 1 0 0 1 0 0 0 0 1 1 1 1

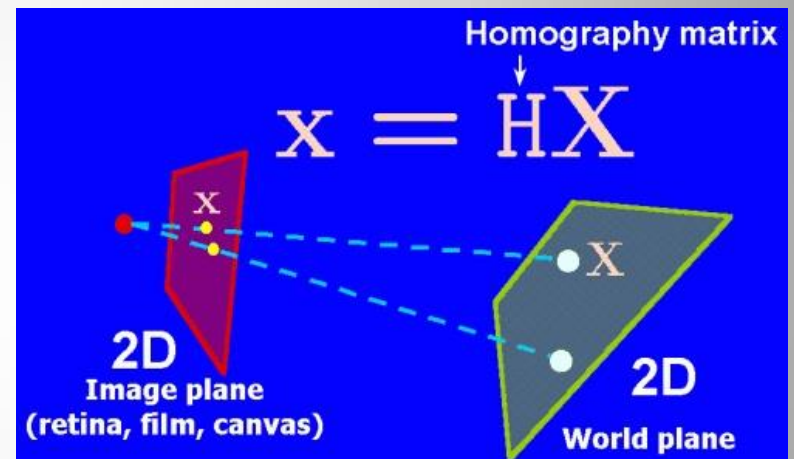
Distance de Hamming = 3

- Si on a un nombre de coins appariées suffisants, l'objet est retrouvé

Relocalisation 2D du pattern

Calcul de l'homographie du plan

- Système d'équation linéaire
- Estimation robuste (RANSAC)
- Filtrage des outliers
- Décomposition en VP

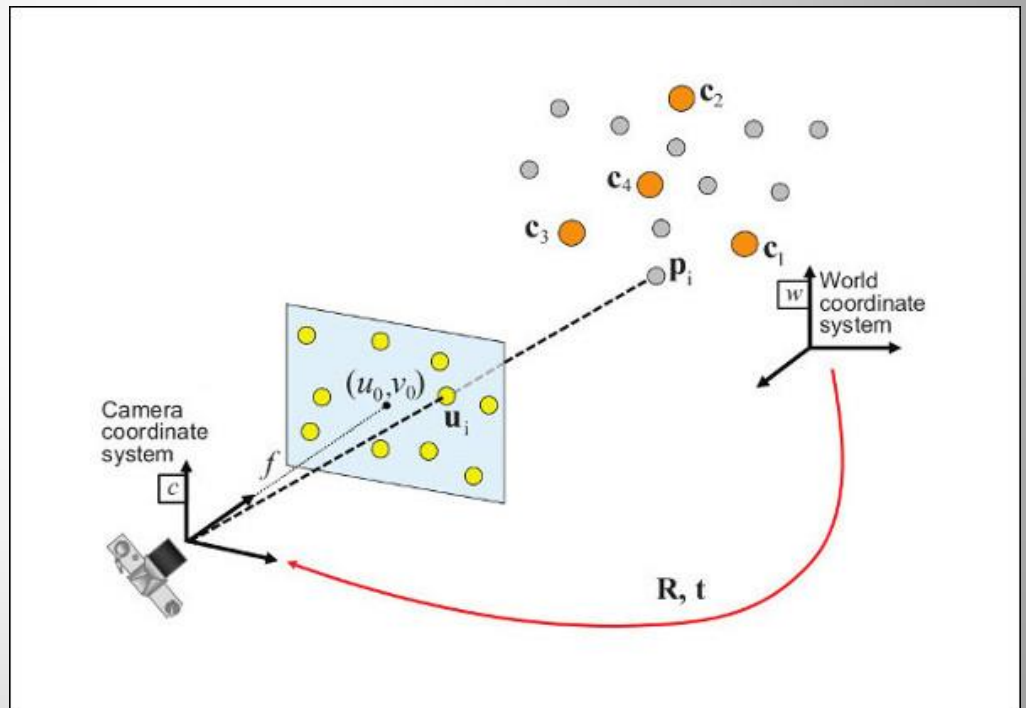


$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_{\text{homography } H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Cacul de la Pose 3D

Calcul de la pose de la caméra par rapport à un objet 3D

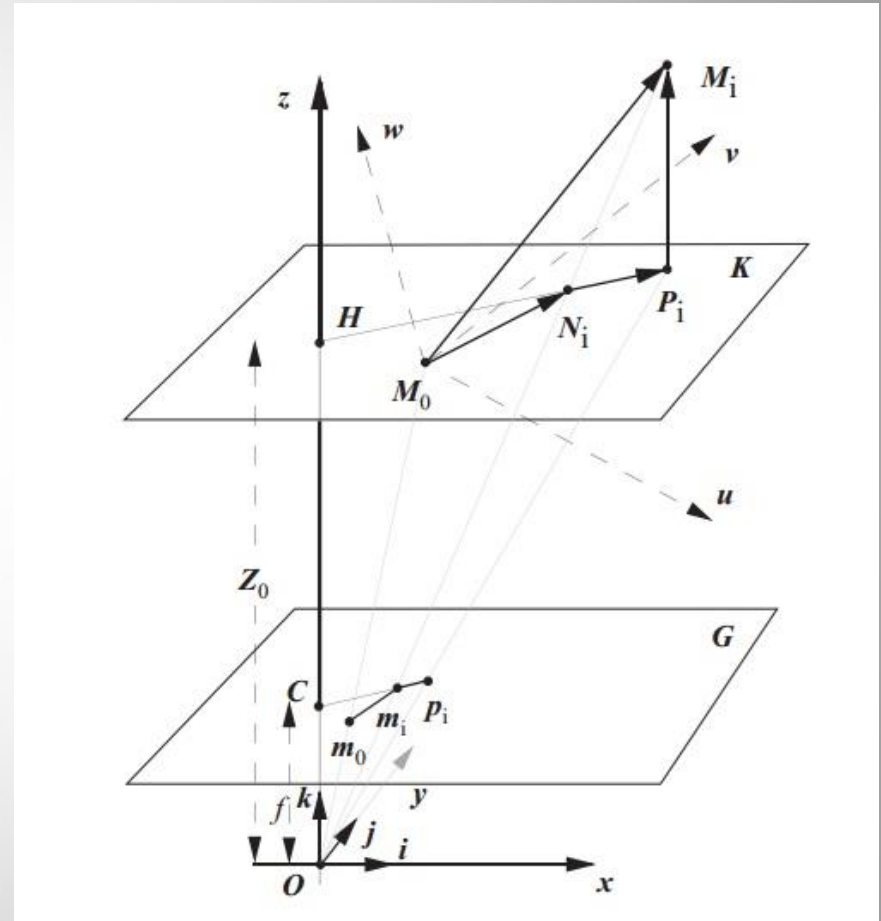
- General case:
 - 6DoF
 - Projection model
- Simplification
 - Calibration connue
 - Perspective-n-Point
 - Projection ortho
 - POSIT



POSIT

POSIT: **P**ose from **O**rthography and **S**caling with **I**terations

- Algorithme itératif pour résoudre PnP non coplanaires
- 4 points coplanaires:
Coplanar POSIT



More on Pose 3D

Calcul de la pose de la caméra par rapport à un objet 3D

- POSIT: [original publications](#), [3D pose estimation](#)
- [Real Time pose estimation](#) : OpenCV tutorial, C++
- [Eric Marchand](#): Article Complet Pose 3D AR
- [Caméra calibration](#): OpenCV tutorial, C++
- [posest](#): C++ opensource
- [Minimal problems](#) in Computer Vision: many links
- Moving camera = Kalman/SLAM

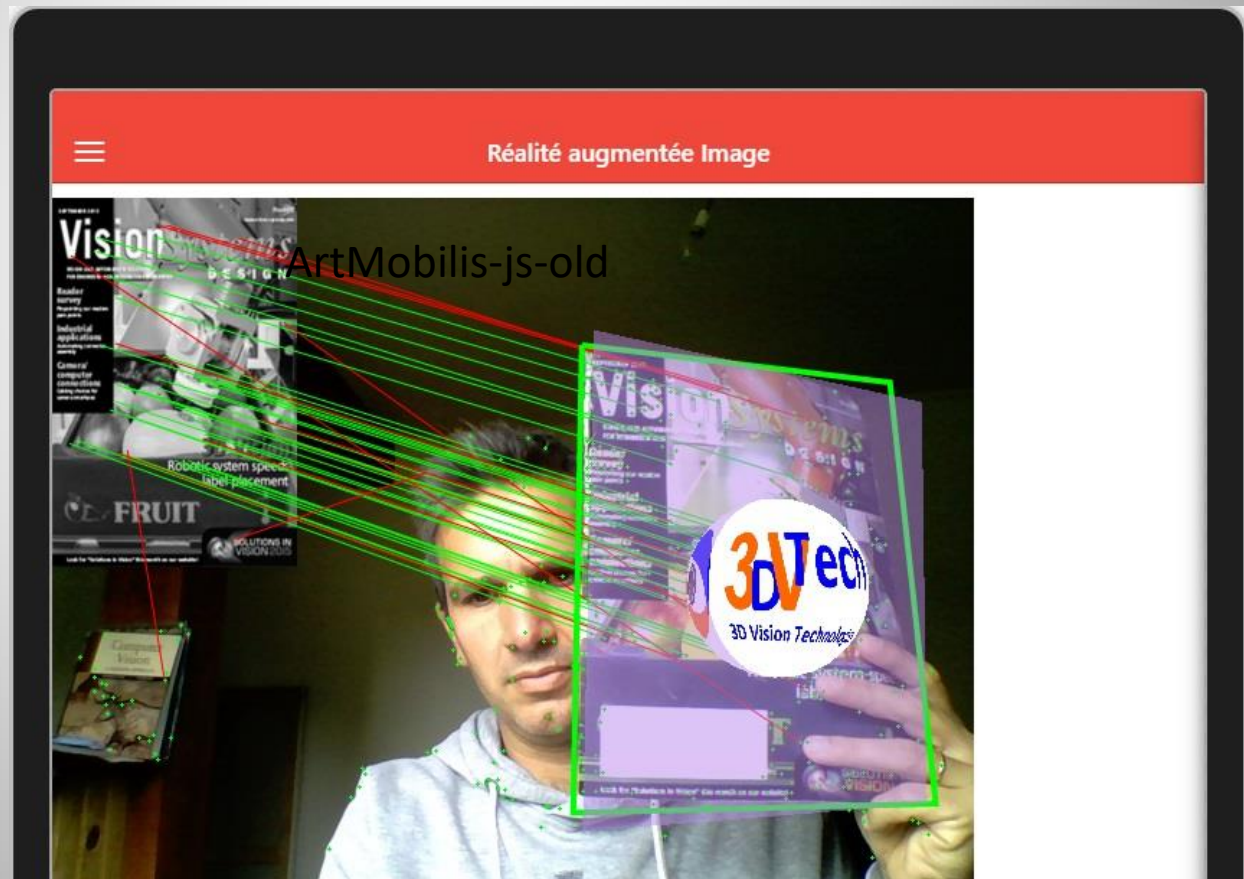
Objectif de ArtMobilis

Un parcours urbain en réalité augmentée

- Géolocalisation des points d'intérêts
- Tracking de la localisation des contenus augmentés
- Support mobile (android, IOS, tablettes)
- OpenSource: <https://github.com/artmobilis/>
- LabMobilis:
 - Implémentation orientée Web pour adaptabilité
 - Application HTML5, CSS3 et JavaScript

Prototype développé

- Demo ArtMobilis-js-old
- Code



Navigateurs compatibles

- [CanIuse](#): 67% des navigateurs
- Compatible avec Firefox/chrome/AndroidBrowser/Edge

getUserMedia/Stream API 📄 - WD

Global 9.66% + 57.74% = 67.4%
unprefixed: 1.03%

Method of accessing external device data (such as a webcam video stream). Formerly this was envisioned as the <device> element.

Current aligned		Usage relative		Show all							
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android		
8			1 45					4.3			
9			1 46					4.4			
10		43	1 47			8.4		4.4.4			
11	13	44	1 48	9	1 34	9.2	8	1 47	1 47		
	14	45	1 49	9.1	1 35	9.3					
		46	1 50		1 36						
		47	1 51								

Librairies Javascript utilisées

- **Framework:**

- Angularjs
- Ionic
- Cordova

- **AR Image demo:**

- Js-ArUco: <https://github.com/jcmellado/js-aruco>
- three.js : <https://github.com/mrdoob/three.js>
- jsfeat : <https://github.com/inspirit/jsfeat>

Exercices

- **Chrome (https only):**

- Bloque getUserMedia pour les fichiers locaux
- Lancer avec --disable-web-security pour du debug
- Navigator.getUserMedia plus supporté -> MediaDevices.getUserMedia()
- Il faudrait utiliser adapter.js
- Attention: exemples pas mis à jour -> utilisez Firefox

- **Firefox:**

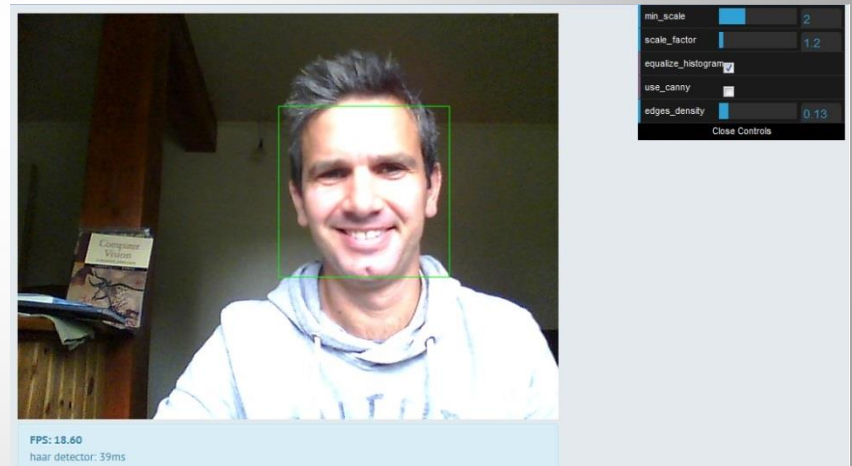
- Version 40 et +: pb avec les vieilles cartes graphique blacklistées
- Installer version 31 pour du debug (marche sur mon laptop)

Exercices

- <https://github.com/vestri/CoursAR>
- **Forkez le repository sous github**
- **Téléchargez le Code**

Jsfeat

- [Jsfeat](#): JavaScript Computer Vision library
- Algorithmes modernes de vision pour Html5
 - Custom data structures
 - Basic image processing
 - Linear Algebra and Multiview
 - Feature 2D
 - Optical flow
 - Object detection



Jsfeat Exercice

- Code dans testhtml/ImageProcessingJSfeat
 - Mettre l'image en **noir et blanc**
-
- **Exercice d'évaluation**

Jsfeat Solution

```
// process each acquired image
function tick() {
    compatibility.requestAnimationFrame(tick);
    stat.new_frame();
    if (video.readyState === video.HAVE_ENOUGH_DATA) {
        ctx.drawImage(video, 0, 0, 640, 480);
        var imageData = ctx.getImageData(0, 0, 640, 480);

        // greyscale conversion
        stat.start("grayscale");
        // I should put my code here
        jsfeat.imgproc.grayscale(imageData.data, 640, 480, img_u8);
        stat.stop("grayscale");

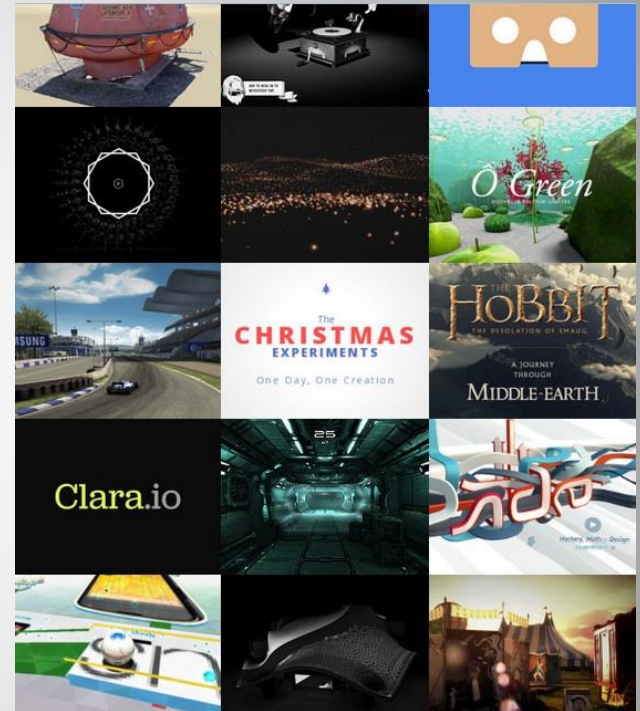
        // render result back to canvas (Warning: format is RGBA)
        stat.start("rewrite");
        // I should put my code here
        var data_u32 = new Uint32Array(imageData.data.buffer);
        var alpha = (0xff << 24); // opacity=1
        var i = img_u8.cols * img_u8.rows, pix = 0;
        while (--i >= 0) {
            pix = img_u8.data[i];
            // write 4 channels: RGBA with GreyGreyGreyAlpha
            data_u32[i] = alpha | (pix << 16) | (pix << 8) | pix;
        }
        stat.stop("rewrite");

        ctx.putImageData(imageData, 0, 0);
        log.innerHTML = stat.log();
    }
}
```

Three.js

[Three.js](#) simplifie l'utilisation de WebGL

- Renderers: WebGL, <canvas>, <svg>...
- Scenes, Cameras, Geometry, Lights, Materials, Shaders, Particles, Animation, Math Utilities
- Loaders: Json compatible Blender, 3D max, Wavefront OBJ



Three.js Exercice

- Code dans testhtml/ImageProcessingThreeJS
- Combiner **l'image et la 3D**

Three.js Exercice

- Code dans testhtml/ImageProcessingThreeJS
- Combiner **l'image et la 3D**
- Indice: Utiliser des Layers

Three.js Solution

```
<title>JSFeat-ThreeJS - Exercice: Canvas 2D 3D</title>
<h1>JSFeat-ThreeJS - Exercice: Canvas 2D 3D</h1>

<video id="webcam" style="display:none;" height="480" width="640"></video>
<div style="width:640px;height:480px;margin: 10px auto;">
  <canvas id="canvas2d" width="640" height="480" class="overlapcanvas"></canvas>
  <canvas id="canvas3d" width="640" height="480" class="overlapcanvas"></canvas>
  <div id="log"></div>
</div>
```

```
.overlapcanvas{
  width:640;
  height:480;
  position: absolute;
  float: left;
  top: 0px;
  left: 0px;
}
```

```
function createRenderersScene() {
  renderer3d = new THREE.WebGLRenderer({ canvas: canvas3D, alpha: true });
  renderer3d.setClearColor(0xffffff, 0);
  renderer3d.setSize(canvas2d.width, canvas2d.height);

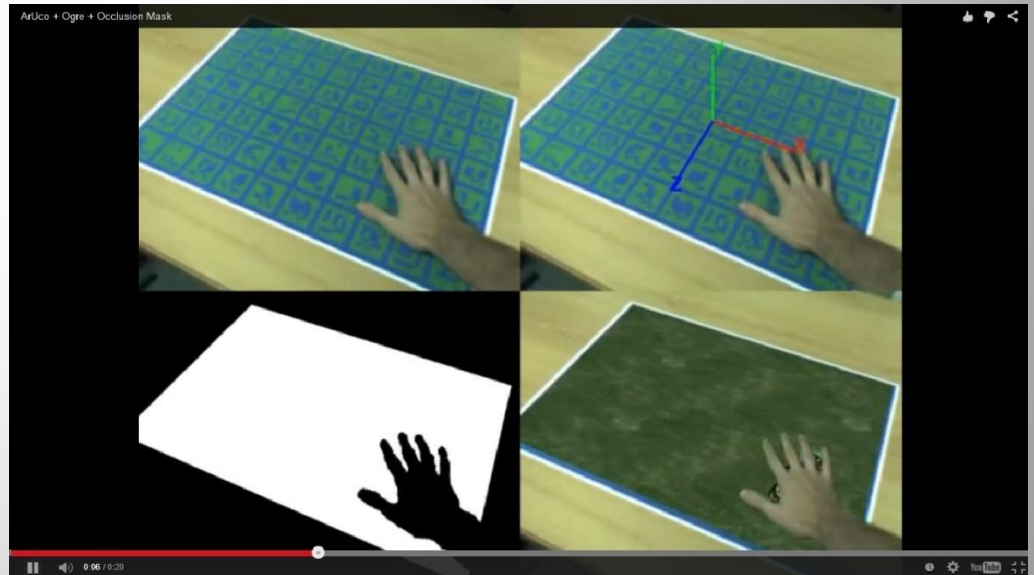
  // for 3d projection
  scene = new THREE.Scene();
  camera = new THREE.PerspectiveCamera(40, canvas2d.width / canvas2d.height, 1, 1000);
  scene.add(camera);

  model = createModel();
  scene.add(model);

  camera.position.z = 5;
};
```

Aruco

- [ArUco](#) est une librairie minimale pour la Réalité Augmentée à base de marqueurs (basée OpenCV)
- [js-aruco](#) est le portage en JavaScript d'ArUco
 - Image processing
 - Contours
 - Detection marqueurs
 - Calcul de pose



ArUco Exercice

- Code dans ImageProcessingAruco
- Faire **tourner la sphère**

ArUco Solution

```
stat.start("Posit");  
pose = posit.pose(corners);  
stat.stop("Posit");  
  
stat.start("Update");  
updateObject(model, pose.bestRotation, pose.bestTranslation);  
stat.stop("Update");  
  
step += 0.025;  
model.rotation.z -= step;  
    }  
};
```


Autres Exercices

- Ajouter un autre objet dans demo Aruco
- Ajouter des effets sur l'image dans demo JsFeat
 - Contour
 - Couleur inversée
 - Heatmap
 - Transparency...

Autres librairies intéressantes

- Computer Vision:
 - Opencvjs: https://docs.opencv.org/master/d0/d84/tutorial_js_usage.html
<https://huningxin.github.io/opencv.js/samples/>
 - tracking.js: <https://github.com/eduardolundgren/tracking.js>
 - js-objectdetect: <https://github.com/mtschirs/js-objectdetect>
 - Convnetjs: <https://github.com/karpathy/convnetjs>
 - sgdSlam: <https://github.com/odestcj/sgd-slam>
- 3D: Babylon.js: <https://github.com/BabylonJS/Babylon.js>
- AR: <http://argonjs.io/> <https://awe.media/>

Projet final cours AR

- Objectifs:
 - 1 projet chacun avec AR inside
 - Outil que vous voulez: Unity, Vuforia, JS, Arcore, Arkit...
 - Présentation le dernier cours
- Planning
 - Trouver un sujet/idée en RA pour la semaine prochaine
 - Unity/vuforia cette semaine, JavaScript semaine prochaine

Vidéo prochain exercice

- <https://www.youtube.com/watch?v=bRZQn4AcqZg>



Exercices

- Ajouter des objets fixes dans la scène
- Fond sonore (musique star wars par ex)
- Faire voler un Xwing ou autre
- Mettre une video de maitre Yoda dans cockpit
- Tester sur votre mobile si Android SDK (ios)
- Surprenez-nous...

Pour la prochaine fois

- **Commencez ou Continuez votre projet**

Plus d'infos

- **Réalité Augmentée:**
 - RAPRO: <http://www.augmented-reality.fr/>
 - SDK liste: [Social Compare-AR-Sdk](#)
 - Lunettes RA: [Social Compare-AR-lunettes](#)
- **Projet**
 - <https://github.com/artmobilis/>
 - vestri@3DVTech.com